HRD Database Evaluation -- July 1998

HRD's Hurricane Wind Analysis System (H*WIND) needs:

H*WIND, formerly Wind Analysis Distributed Application (WANDA), is a critical software a critical software application, a real-time analysis system that provides the most current diagnosis information on the actual areas impacted by hurricane fields. At present, H*WIND, although an object-oriented application, accesses a file-based database system composed of flat ASCII and binary files stored in several directories. The principal disadvantages of such a system are:

- Duplication of data items - H*WIND's need to work with subsets of data sets requires that the same data be stored in several files. The analysis subsystem, a component of the H*WIND application, works on observation data that undergoes quality control. The same data is duplicated in several files each time the user produces an analysis.
- Application dependency - The file layout depends on the H*WIND application. Only H*WIND can access and manipulate the data.
- Inflexibility - This works hand in hand with application dependency where a change in the file layout requires an application change.

The trend is overwhelming in favor of replacing file-based systems and applications with database management systems (DBMS) and applications. In a database environment, applications are built around an integrated database. The principal advantages of a database are the ability to share the same data across multiple applications and systems, store data in flexible formats, and allow the use of data (data independence) independently of the users and applications. DBMSs have evolved through a number of generations including hierarchical, network, relational, object-oriented, semantic, multidatabase systems, and object-relational systems. The relational model remains, however, as the dominant database technology.

There is an increasing need to store and manipulate complex data in relational database systems. Complex data is imminent not only in multimedia applications for the Web, but also in specialized application domains such as meteorology, geographical, space, medical, and exploration systems (such as maps, radar data, satellite images, and etc.). For example, Binary Large Objects (BLOB), spatial and time series data, user-defined-datatypes (UDTs) (datatypes that may encapsulate complex internal structures and attributes), and user-defined-functions (UDFs) (methods by which applications can create, manipulate, and access the data stored in these new datatypes) cannot be easily stored in relational databases.

Our goals and prerequisites are as follows:
(See, Table 1,2, and 3 for comparison analysis of database systems).

- A database should provide for the efficient storage, update, and retrieval of non-complex and

complex (BLOB, UDTs, UDFs, etc.) data.

- A database should support data and method encapsulation (users are not required to understand the internal data structures nor the methods that manipulate them) and overloading (the ability to use the same name for different routines). For example, the function GetQcObs() allows the retrieval of all quality control observations such as land, oceanic, stormfixes, etc.

Database Systems	Complete Database Management System	Storage Efficiency Data	Storage Efficiency for Complex Data
OpenBase	Y	Y	N
NETCDF	N	N	Y
Informix	Y	Y	Y
Oracle 7.x	Y	Y	N
OSMOS	Y	Y	Y
Gemstone	Y	Y	Y
Oracle 8	**Y**	**Y**	**Y**

Table 1: DBMS Storage Efficiency

Database Systems	Data Encapsulation	Availability by July 1997	Java Compatible by middle 1997
OpenBase	N	Y	N
NETCDF	N	Y	N
Informix	Y	N	N
Oracle 7.x	N	Y	Y
OSMOS	Y	N	N
Gemstone	Y	Y	N
Oracle 8	**Y**	**Y**	**Y**

Table 2: DBMS Data Encapsulation, Availability, Java Compatibility

Database Systems	Reliability	License Cost	Training Availability: 0 = none, 1 = vendor/owner, 2 = many vendors
OpenBase	Y	FREE	0
NETCDF	Y	FREE	0

Informix	Y	+$19,000/5 USER	1
Oracle 7.x	Y	$1,450/5 USER	2
OSMOS	Y	NA mid 1997	0
Gemstone	Y	+$21,000/5 USER	1
Oracle 8	**Y**	**$1,450/5 USER**	**2**

Table 3: DBMS Reliability, License Cost, Training Availability

- A database should be reliable - the stored data should have high integrity to promote user trust in that data.
- A database should be adaptable and scaleable to new and unforeseen requirements and applications.

Based on our needs, HRD conducted an independent evaluation of the following database systems:

OpenBase Relational (RDBMS)

The Structured Query Language (SQL) is a database query language that has become the industry standard to retrieve and manage data, more specifically the relational data. A major revision of SQL (SQL89) came in 1992, SQL92. Another SQL revision, SQL3 is scheduled for the end of 1998. The leading RDBMS vendors, ORACLE, IBM, INFORMIX, and SYBASE, are using SQL92 and have already included new enhancements promised from SQL3. OpenBase's native language is SQL-89 embedded in Objective-C. RDBMSs are well know for their implementation of integrity constraints residing inside the database server. Integrity constraints protect user data at the level of the database engine versus at the programmer's level. OpenBase offers only basic (limited) integrity constraints. Database vendors must provide means through which to communicate and access databases from other vendors. OpenBase provides no information on compatibility or connectivity with other database systems.

NETCDF SOFTWARE

NetCDF (network Common Data Form) is an interface for array-oriented data access and a freely-distributed collection of software libraries for C, Fortran, C++, and perl that provide implementations of the interface. The sophistication of gridded file input, multidimensional arrays cannot be easily stored in relational database. AWIPS uses NetCDF for storage & retrieval of these complex datatypes. AWIPS's employees indicated to AOML's database developers that NetCDF although excellent for

multidimensional arrays, was inefficient for storing other datatypes, such as text data. Furthermore, NetCDF is not a database management system. Data manipulation remains at the level of the programmer. NetCDF is used in some enterprises as a complement to commercial relational database systems. The use of one database systems to fulfill the needs of an application is preferred.

ORACLE RELATIONAL DATABASE SYSTEM (version 7.3x)

Relational database technology is widely deployed and used in contemporary information system hops. However, the principal drawback of the relational database management systems (RDBMS) is their inability to store and manipulate complex data, in particular user-defined-datatype (UDTs) and user-defined-functions (UDFs). Some RDBMS's vendors have added, so called "object extensions" or "plug-ins" to the database engine in an effort to accommodate the industry wide for the manipulation of complex data. For example, SYBASE has added "object extensions" to its relational database engine. H*WIND was written first in Objective-C (Object extensions added to the C language) and has been redesigned using JAVA. H*WIND is an object-oriented application. H*WIND needs to use database systems that can accommodate the objects that the application produces. Therefore, relational database systems are not a viable choice for this application.

GEMSTONE OBJECT-ORIENTED DATABASE SYSTEM (OODBMS)

OODBMS provide persistent storage for object-oriented programming languages (OOPLs), such as Smalltalk or C++. Application design tools, such as CAD and CASE (CASE and CAD, engineering drawing case tools) have been instrumental in the evolution of object database technology. Typically, applications in these environments load a large number of objects into local memory (such as a circuit board design or an automobile design), manipulate them extensively (sometimes for hours or even days), and then store them back into the database. To facilitate this type of processing, persistent object oriented programming language databases typically support specific design-oriented activities, such as object check-in/check-out, object versioning, and long transactions.

The language used in the database engine of database systems is called the "native" language of the system. GEMSTONE's native language is SMALLTALK. One of the disadvantages of the OODBMS systems is that it requires the knowledge of those object-oriented programming languages. The learning curve for these languages is quite long in comparison to learning SQL. SMALLTALK is an object-oriented programming language which is known only in the object-oriented world which, though growing, remains a limited circle. Therefore, training and support for Smalltalk applications are hard to combine. Additional disadvantages are: lack of GUI tools, database administrators, database designers, and database developers on the market.

OSMOS (ORDBMS)

Object-Relational database systems (ORDBMS) attempt to combine the best features from RDBMS and OODBMS technology in a single product. Object-relational databases are good for storing complex data types or complex relationships between the data. Complex relationships are typically found in scientific, risk management, time-series analysis applications, etc.

ORDBMS applications tend to be simpler than comparable relational database applications due to the increased responsibilities assumed by the database. ORDBMSs provide high-level constructs, such as abstract data types, hierarchies, and bi-directional relationships, allowing a database schema to be expressed in the real-world view of the business problem being solved. This increases overall data integrity and increases the domain complexity in which it is practical to implement application solutions.

OSMOS is a high-performance, object-relational database created by Unisys Corporation. You can use OSMOS to create applications such as point-of-sale transactions, catalog ordering (especially online), financial applications, any applications requiring multimedia elements, and applications that need large amounts of data available to many concurrent users. At the time that we investigated OSMOS, only BellSouth was using the system and were unable to get any technical support from Unisys while evaluating the software.

INFORMIX UNIVERSAL SERVER

The Informix Universal Server was unavailable at the time of our investigation. We were unable, without commitment to purchase the database software, to test even the beta version of the software. However, we tested ILLUSTRA (from Stonebaker), an object-relational database. Informix merged some features from ILLUSTRA with its relational database and called it Informix Universal Server. The Informix Universal Server was scheduled to be released at the beginning of 1997. The first version was released in September 1997 with a price tag of approximately $ 10, 000 per user license and only for the UNIX platform. Oracle 8 for NT was released in July 1997 and cost $1,450 per 5 user license. In July 1997, Oracle Corporation sent us the Oracle 8 software, with no commitment and with free technical support for four months. The cost of the database software and hardware, technical support availability, the uncertainty of what features of ILLUSTRA that would be merged with Informix relational database, and the fact that the Informix Universal Server was not ready in July 1997 were the important factors that supported our decision to purchase Oracle 8.

ORACLE 8 OBJECT-RELATIONAL DATABASE SYSTEM (ORDBMS)

ORACLE 8 Object-Relational Database adds new object storage capabilities to the relational systems. These new facilities integrate management of traditional fielded data, complex objects such as time-series and geospatial data and diverse binary media such as audio, video, images, and applets. By encapsulating methods with data structure, ORACLE 8 can execute complex analytical and data manipulation operations to search and transform complex objects. It has inherited the robust transaction

and performance management features of its relational ancestor ORACLE 7.x and the flexibility from some of the object-oriented concepts. Database designers can work with easy and popular structures of the Structured Query Language (SQL) and data definition languages (DDL) while assimilating new object-management possibilities. The most important new object-relational features are user-defined-datatypes (UDTs), user-defined-functions (UDFs), and the infrastructures - indexing/access methods and optimizer enhancements - that support them.

ORACLE 8 was chosen based on advantages such as:

- Software installation - Oracle 8 installation does not require the presence of the vendor. For example, INFORMIX customers are required to hire a database administrator from INFORMIX Corporation and pay hundreds of dollars to install the software. Oracle 8 was installed by HRD's staff and a sample database was up and running in less than one hour.

- Manageability - Creating and backing up the database, configuring database files, monitoring the database instance (when the database is up and running, it is referred as a database instance) and user sessions are not complicated tasks. Oracle provides the GUI tools to help database administrators and developers to perform these tasks.

- Performance - According to Oracle, data retrieval takes ~ 1micro second on complex and non-complex data.

- Scalability - Oracle 8 can accommodate thousands of concurrent users simultaneously. There is no limitation - limited by hardware - on the number of users that can concurrently use the software.

- Technical support - A phone call away and the creation of a problem number (Tar identification number). In general, Oracle responds within 24 hours after the generation of a tar number.

- GUI tools and documentation - In July 1997, GUI tools and documentation on Oracle 8 were non existent. Today, most libraries and bookstores have books and articles on Oracle 8.

- Training - Most training centers offer training on Oracle 8 database administration, installation, design, and implementation. Training on Informix Universal Server is provided only by Informix Corporation.

The above advantages were determinant factors that supported our decision to purchase Oracle 8 rather than another database management system.

Funded by NOAA High Performance Computing and Communications (HPCC)